

3 a.**3.a.i.**

This program was made on the code.org website to provide a game that will help you recognize new words and expand your vocabulary.

3.a.ii.

In the video it shows the user inputting many letters and the program either outputting the letter where the letter belongs in the randomly generated word or taking a life from the user since the letter was not found in the word.

3.a.iii.

The video shows the user inputting the letter "p" and the program outputting the letter in the spot that it belongs in the word (ex: Letter Inp: P --> Output: p_ _ _ _). The user also inputs numbers, symbols, multiple letters in a single guess, and letters that do not belong in the word. In the first game of the video, the user guesses all the letters in the words and gets prompted with a winning screen. In the second video the user runs out of lives and gets prompted with a losing screen.

3 b.**3.b.i.**

```

32 function getRandomWord(){
33     count = 0;
34     lives = 6;
35     setText("livesCount", lives);
36     for(var i = 0; i < allWords.length; i++)
37         if(allWords[i].length == 5)
38             addItem(words5, allWords[i]);
39     guessingWord = words5[randomNumber(0, words5.length - 1)];
40     for(var j = 0; j < guessingWord.length; j++)
41         letOfGuessWord[j] = guessingWord.substring(0+j, j+1);
42     for(var x = 0; x < letOfGuessWord.length; x++){
43         setText("let" + x, "");
44     }
45 }

```

3.b.ii.

```

48 ▾ for(var i = 0; i < letOfGuessWord.length; i ++){
49 ▾     if(letter == letOfGuessWord[i]){
50         setText("let" + i, letter);
51         check = true;
52         count = count + 1;
53     }
54 }

```

3.b.iii.

"letOfGuessWord"

3.b.iv.

The data that my list contains is vital to my program since it holds the individual letters of the word that the user is trying to guess in their own individual indexes.

3.b.v.

If the program were to run without the "letOfGuessWord" list, it would make everything very inefficient because I would have to make 5 new variables to store the individual letters, that is taken from the randomly generated word, into each variable. This will also make my for loop more complicated since I would have to loop through the strings to see if the letter that the user inputted, is in fact in the word.

3 c.

3.c.i.

```

46 function guessWords(letter) {
47     check = false;
48     for(var i = 0; i < letOfGuessWord.length; i ++){
49         if(letter == letOfGuessWord[i]){
50             setText("let" + i, letter);
51             check = true;
52             count = count + 1;
53         }
54     }
55     if(check == false){
56         lives = lives - 1;
57         setText("livesCount", lives);
58     }
59     if(lives == 0){
60         setScreen("losingScreen");
61         setText("finWord", "This was the word: " + guessingWord);
62     }
63     if(count == letOfGuessWord.length)
64         setScreen("winningScreen");
65     return check;
66 }

```

3.c.ii.

```

13 onEvent("textInp", "change", function( ) {
14     var inpCase = getText("textInp");
15     guessWords(inpCase.toLowerCase());
16     setText("textInp", "");
17 });

```

3.c.iii.

The function does most the work by basically comparing the letters of the randomly generated word to the letter that the user has inputted. If the user has gotten it correct, the letter will pop up in the space where it belongs. If the program does not find the letter in the word, it will take a life away from the user.

3.c.iv.

I have a Boolean, "check", that is set to false, a variable "count" that is set to 0, and a variable "lives" that is set to 6. Then I use a for loop that loops through the entire list by the length of the word that the user is trying to guess. Inside the for loop I have an if statement that compares the letter that the user inputted, with all of the letters in the randomly generated word. If the if statement condition is true it does the following: it sets the letter that the user inputted to let x, x being the index of where the letter was found in the randomly generated word, then it will change "check" to true since it did indeed find a letter in the word, lastly it will add 1 to "count" since it needs to keep track of how many times the user has guessed a letter correctly. Then it has 3 extra if statements outside of the for loop to clean up everything. The first if statements checks if "check" is false, meaning the user did not guess a letter in a word, and if it is it takes a live away from the user. Second if statement checks if the user has 0 lives and if the user does indeed have 0 lives, the user will be prompted with a losing screen. Third if statement checks if "count" is equal to the word length that the user is trying to guess, meaning that the user has guessed all of the letters in the word, and if it is it shows the user the winning screen. Lastly I would return check to see if the letter that the user inputted was in the word or not.

3 d.

3.d.i.

First call:

First example will be when the user inputs the number "1" and the word that the user is trying to guess being "hello".

Second call:

Second example will be when the user inputs the letter "h" and the word that the user is trying to guess being "hello".

3 d.ii.

Condition(s) tested by first call:

In line 49, the if statement compares the number "1" to all of the letters in "hello".

Condition(s) tested by second call:

In line 49, the if statement compares the letter "h" to all of the letters in "hello".

3.d.iii.

Results of the first call:

The result of the first call is false due to the number "1" not being found in "hello"

Results of the second call:

The result of the second call is true due to the letter "h" being found in "hello"